

Remarks

This Application has been carefully reviewed in light of the Office Action mailed November 2, 2007. Applicant appreciates the Examiner's consideration of the Application. Applicant believes all claims are allowable without amendment and respectfully provides the following remarks. Applicant respectfully requests reconsideration and allowance of all pending claims.

I. The Claims are Allowable over *Ankireddipally*

The Examiner rejects Claims 1-30 under 35 U.S.C. § 102(a) as being anticipated by U.S. Patent 6,971,096 to Ankireddipally et al. ("*Ankireddipally*").¹ Applicant respectfully disagrees.

"A claim is anticipated only if *each and every element as set forth in the claim* is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 U.S.P.Q.2d 1051, 1053 (Fed. Cir. 1987) (emphasis added); M.P.E.P. ch. 2131. In addition, "[t]he identical invention must be shown in *as complete detail as contained in the . . . claim*." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 U.S.P.Q.2d 1913, 1920 (Fed. Cir. 1989) (emphasis added); *see also* M.P.E.P. ch. 2131. Furthermore, "[t]he elements must be arranged *as in the claim* under review." *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990) (emphasis added); M.P.E.P. ch. 2131. As illustrated below, the cited portions of *Ankireddipally* do not appear to disclose, either expressly or inherently, each and every limitation recited in Applicant's independent claims.

A. Independent Claims 1 and 8 and their Dependent Claims are Allowable

The cited portions of *Ankireddipally* do not appear to disclose, teach, or suggest at least the following limitations recited in Claim 1 (which Applicant discusses as an example):

- pars[ing] a plurality of transaction definitions for a software system, wherein each transaction definition comprises one or more parameters; and

¹ The Office Action actually lists US Patent Publication No. 2003/0172368 A1. However, it appears to Applicant that this is a mistake. The reference cited in the previous Office Action (i.e., *Alumbaugh*) corresponds to that number. Moreover, the Examiner's cites to *Ankireddipally* in the current Office Action appear to correspond to U.S. Patent 6,971,096.

- generat[ing] a plurality of schema definitions in response to the plurality of transaction definitions, wherein the schema definitions are written in a self-describing language.

As allegedly disclosing these limitations, the Examiner cites the Abstract; Column 7, lines 30-45; and Column 12, lines 29-41 of *Ankireddipally*. (Office Action at 3) The Abstract discloses that a process automation application (a CX server) for sending transaction messages between application programs uses a transaction definition data structure for specifying the component operations and processing logic that comprise the transaction. The Abstract discloses that the transaction definition data structure may be an XML document in the form of a directed acyclic graph. While the Abstract discloses a transaction definition data structure that is an XML document, the Abstract does not appear to disclose, teach, or suggest “pars[ing] a plurality of transaction definitions for a software system, wherein each transaction definition comprises one or more parameters,” and “generat[ing] a plurality of schema definitions in response to the plurality of transaction definitions, wherein the schema definitions are written in a self-describing language,” as recited in Claim 1.

Column 7, lines 30-45 of *Ankireddipally* state the following:

Therefore, in accordance with one aspect of the present invention, there is provided an XML (extensible markup language) transaction definition document stored on a computer-readable medium comprising a plurality of operation data portions each defining an operation. The plurality of operations collectively define a transaction. Each operation data portion, when parsed by a process automation application, causes the process automation application to communicate with a service application program to perform the operation. At least one operation data portion comprises a conditional logic data portion that, when parsed by the process automation application, causes the process automation application to condition performance of a next operation on evaluation of operation response data from performing the operation.

While this cited portion discloses “parsing,” the parsing is of the transaction definition document, which is the item disclosed in *Ankireddipally* that Applicant believes the Examiner is attempting to equate to the documents written in the self-describing language recited later in Claim 1.² The transaction definition document disclosed in *Ankireddipally* cannot be equated with both the thing parsed in Claim 1 (i.e., the transaction definitions for a software

² If Applicant’s interpretation of the Examiner’s position is inaccurate, Applicant respectfully requests that the Examiner clarify his position in the next Office Action (should there be a next Office Action).

system) and the things generated (i.e., either or both of the first and second documents written in the self-describing language) based the schema definitions generated in response to the parsed transaction definitions.

Column 12, lines 29-41 of *Ankireddipally* state:

Persistence service 19 provides interfaces for storing information into and retrieving information from external data stores 18. From the perspective of CX server 10 or a CXC, data entering into or coming from data stores 18 are in XML document format. Persistence service 19 has the responsibility of mapping between an XML document and the respective data store schema. In an illustrated implementation of CX server 10, data stores 18 include a Netscape.TM. message server, a Netscape.TM. LDAP server, and an Oracle.TM. database server. Support for flat files is also possible. Examples of information that are included in data stores 18 are system parameters, events and alerts, and transaction definitions.

Among other things, the cited portion discloses that persistence service 19 maps an XML document to a data store schema. However, it does not appear to Applicant that the cited portion discloses, teaches, or suggests “generat[ing] a plurality of schema definitions *in response to the plurality of transaction definitions* [parsed in the previous limitation of Claim 1], wherein the schema definitions are written in a self-describing language,” as recited in Claim 1.

Furthermore, the cited portion appears to have no clear relation to the portion of *Ankireddipally* the Examiner cited against “pars[ing] a plurality of transaction definitions for a software system, wherein each transaction definition comprises one or more parameters,” as recited in Claim 1. Therefore, it is not clear how this cited portions could possibly disclose, teach, or suggest “generat[ing] a plurality of schema definitions *in response to the plurality of transaction definitions [supposedly disclosed in the above-cited portions of Ankireddipally]*, wherein the schema definitions are written in a self-describing language,” as recited in Claim 1.

For at least these reasons, Applicant respectfully requests reconsideration and allowance of independent Claim 1 and its dependent claims. For at least certain analogous

reasons, Applicant respectfully requests reconsideration and allowance of independent Claim 8 and its dependent claims.

B. Independent Claims 12 and 22 and their Dependent Claims are Allowable

The cited portions of *Ankireddipally* fail to disclose, teach, or suggest at least the following limitations recited in Claim 12 (which Applicant discusses as an example):

- a software service operable to receive a transaction request and to generate a first object associated with the transaction request;
- an object generator operable to convert the first object into a first document written in a self-describing language; and
- a document generator operable to convert the first document into a first transaction message according to a schema associated with a first transaction type determinable from the first document.

For example, the cited portion of *Ankireddipally* does not appear to disclose, teach, or suggest “a software service operable to receive a transaction request and to generate a first object associated with the transaction request,” as recited in Claim 12. As allegedly disclosing these limitations, the Examiner cites the “CXC” disclosed in *Ankireddipally* at Column 10, lines 3-17. (Office Action at 6) The cited portion states:

Each application component is referred to as a commerce exchange component, or CXC. As shown in FIG. 1, there may be any number of CXCs identified to CX server 10. A CXC application either provides one or more services or originates a transaction request, or both. A CXC application may be integrated with CX server 10 as a built-in component residing on the same machine or it may be a third party application resident on a different machine. For example, service application 30 is resident on machine 24 and accessible to CX server 10 via communications connection 29, and requesting application 38 is resident on machine 36 and accessible to CX server 10 via communications connection 35. The type of architecture model illustrated in FIG. 1 may be variously described in the literature as an information bus model, a client-server model or a cooperative agent model.

(*Ankireddipally* at 10:3-17)

Among other things, this portion of *Ankireddipally* discloses that a CXC application either provides one or more services or originates a transaction request, or both. However, nowhere does the cited portion disclose, teach, or suggest that the CXC is “operable to

receive a transaction request and to *generate a first object associated with the transaction request*,” as recited in Claim 12.

As another example, at least because the cited portions of *Ankireddipally* do not disclose, teach, or suggest “a software service operable to receive a transaction request and to generate a first object associated with the transaction request,” as recited in Claim 12, the cited portions of *Ankireddipally* necessarily fail to disclose, teach, or suggest “an object generator operable to convert the first object into a first document written in a self-describing language,” as recited in Claim 12. As allegedly disclosing the object generator recited in Claim 12, the Examiner cites column 10, lines 32-50 of *Ankireddipally*, referring particularly to the statement “extracting transaction data from XML-based message . . . and returning an XML document.” (Office Action at 6) While the cited portion of *Ankireddipally* discloses “returning an XML document,” it does not appear to Applicant that the cited portion discloses, teaches, or suggests “convert[ing] *the first object* into a first document written in a self-describing language,” as recited in Claim 12.

As another example, at least because the cited portions of *Ankireddipally* does not disclose, teach, or suggest “a software service operable to receive a transaction request and to generate a first object associated with the transaction request” and “an object generator operable to convert the first object into a first document written in a self-describing language,” as recited in Claim 12, the cited portions of *Ankireddipally* necessarily fail to disclose, teach, or suggest “a document generator operable to convert the first document into a first transaction message according to a schema associated with a first transaction type determinable from the first document,” as recited in Claim 12. Moreover, as allegedly disclosing the document generator recited in Claim 12, the Examiner cites column 10, lines 32-50 of *Ankireddipally*, referring particularly to the statements “returns one or more DOM objects . . . for handling as standard program objects.” (Office Action at 6) The cited portion specifically refers to the return of “one or more DOM *objects* . . . for handling as *standard program objects*.” It seems to Applicant that the cited portion plainly does not disclose, teach, or suggest “a document generator operable to *convert the first document into a first transaction message*,” let alone doing so “according to a schema associated with a first transaction type determinable from the first document,” as recited in Claim 12.

For at least these reasons, Applicant respectfully requests reconsideration and allowance of independent Claim 12 and its dependent claims. For at least certain analogous reasons, Applicant respectfully requests reconsideration and allowance of independent Claim 22 and its dependent claims.

II. No Waiver

All of Applicant's arguments and amendments are without prejudice or disclaimer. Additionally, Applicant has merely discussed example distinctions from the reference cited by the Examiner. Other distinctions may exist, and Applicant reserves the right to discuss these additional distinctions in a later Response or on Appeal, if appropriate. By not responding to additional statements made by the Examiner, Applicant does not acquiesce to the Examiner's additional statements. The example distinctions discussed by Applicant are sufficient to overcome the Examiner's rejections.

Conclusion

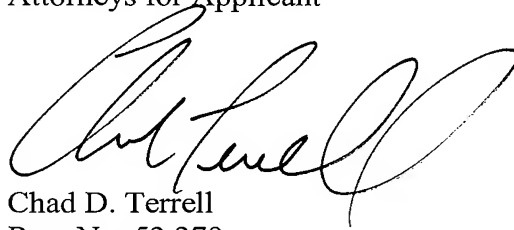
Applicant has made an earnest attempt to place the Application in condition for allowance. For at least the foregoing reasons, Applicant respectfully requests full allowance of all pending claims.

If the Examiner feels that a telephone conference or an interview would advance prosecution of the Application in any way, the Examiner is invited to contact the undersigned attorney for Applicant at the Examiner's convenience at (214) 953-6813.

Although Applicant believes no fees are due, the Commissioner is hereby authorized to charge any necessary fees or credit any overpayment to Deposit Account No. 05-0765 of Electronic Data Systems Corporation.

Respectfully submitted,

BAKER BOTTS L.L.P.
Attorneys for Applicant

A handwritten signature in black ink, appearing to read 'Chad D. Terrell', is written over the printed name and registration number.

Chad D. Terrell
Reg. No. 52,279

Date: February 4, 2008

Customer Number: **35005**